EXHIBIT 16

Smart Cards – Requirements, Properties, and Applications

Klaus Vedder, Franz Weikmann

Giesecke & Devrient GmbH
Prinzregentenstr. 159, D-81677 München, Germany
klaus.vedder@gdm.de, franz.weikmann@gdm.de

Abstract. Smart cards play an increasing role as 'active' security devices. Due to its microcomputer and programmable memory, a smart card can cater for the specific needs of the environment it is used in. Smart cards allow the secure handling and storage of sensitive data such as user privileges and cryptographic keys as well as the execution of cryptographic algorithms. They are secure tokens by means of which a user can be identified and authenticate a computer system or communication network and vice versa. This paper provides a comprehensive introduction into the features of chip cards**,** the principals of their operating system, their life-cycle and the standards governing them. It also includes a brief discussion of major applications and an outlook on the future development.

1 Introduction

"Smart Cards. The ultimate personal computer." is the title of a book by J. Svigals [15] one of the first treatise of this subject. Since its publication in 1985 smart cards have gone a long way towards achieving this claim. What are smart cards? Smart cards are a specific type of chip cards. These are cards, usually made of plastic, containing a chip. Depending on the properties and features of this chip and its 'carrier' we distinguish the following types, which we will discuss briefly to achieve a common understanding of the terminology.

Memory cards contain non-volatile memory and allow 'free' reading and, in many instances, writing or updating of data stored. Writing usually refers to changing a 0-bit/byte to a 1-bit/byte (or vice versa), while updating is erasing the contents of the memory cells followed by writing. Such cards are used instead of magnetic stripe cards as they are more reliable and offer far more memory. Not all types of memory allow the erasure of data, a feature which is a basic requirement in many applications.

Intelligent memory cards contain a security logic in addition to the non-volatile memory. This allows the introduction of security attributes for reading and writing data. A memory zone may be secret (the data is used for card internal purposes only), public or sensitive. The latter means that it is accessible only after the presentation of

B. Preneel, V. Rijmen (Eds.): COSIC'97 Course, LNCS 1528, pp. 307-331, 1998.
Springer-Verlag Berlin Heidelberg 1998

a correct "personal feature" of the user. This is in most cases a Personal Identification Number (PIN) consisting of 4 to 8 digits. The PIN is protected against trial and error attacks by a 'false-presentation-counter'. After a specified number of consecutive false entries the security logic blocks the non-public data against any further access. The new generation of telecommunication chips (for pre-paid telephone applications) also include hardware algorithms for challenge-response mechanisms for the authentication of the card by the system to increase the security against cloning.

Smart cards are chip cards where the chip is a microcomputer with programmable memory.

Super smart cards are smart cards with an integrated key board, a display and solar cells or a battery. Due to its complexity and high price, this type of card has not advanced much beyond field trials.

A *contactless card* can be any of the above. Its name is derived from the way the data is transmitted between the chip and the InterFace Device (IFD) or the Card Accepting Device (CAD); the standardised names for chip card readers. In the last three years the standardisation of such cards has made enormous progress (see chapter 6 below).

Hybrid cards usually refer to chip cards which have two or more interfaces. Such an interface can be a magnetic stripe, contacts, contactless, or an optical memory. Most present-day chip cards use contacts for the transmission of power and data.

2 Interfaces and Dimensions

In this chapter we discuss the interfaces and the dimensions of chip cards or integrated circuit(s) cards (IC cards) with contacts. The dimensions and locations of their contacts are laid down in Part 2 of the International Standard ISO/IEC 7816. This standard [11], which is jointly published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), is the basic reference for such cards.

In addition to the eight contacts, the card can be equipped with a magnetic stripe or be embossed. Embossing and magnetic stripe shall be on opposite sides. The reader is referred to references [9], [10] and [11-2] for details of the specifications of these two features. While the current version of ISO/IEC 7816-2 allows contacts and magnetic stripe to be on the same side, this possibility has been excluded in the revision of this standard which is expected to be published later this year. Figure 1 below thus shows the only possibility for a card which provides all three interfaces. Such cards are typically cobranding cards comprising, for instance, a credit card and a telecommunication function.

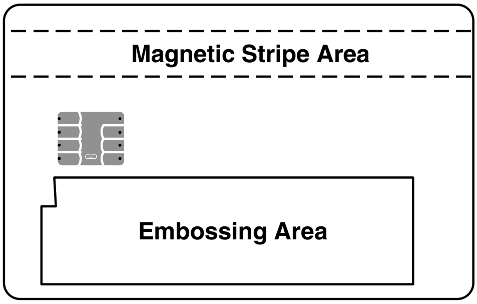


Fig. 1. Interfaces

2.1 Dimensions

The 'standard' identification card or ID-1 card [8] is the size of a credit card. Figure 2 gives the approximate dimensions of this card and the location of contacts as well as height and width of the so called Plug-in card. This ID-000 card with a height of 15 mm and a width of 25 mm has first been specified by the European Telecommunications Standards Institute (ETSI) for GSM, the Global System for Mobile communications, where about half the number of cards have this format [5]. They are mainly used in mobile phones which are too small to accept an ID-1 card.

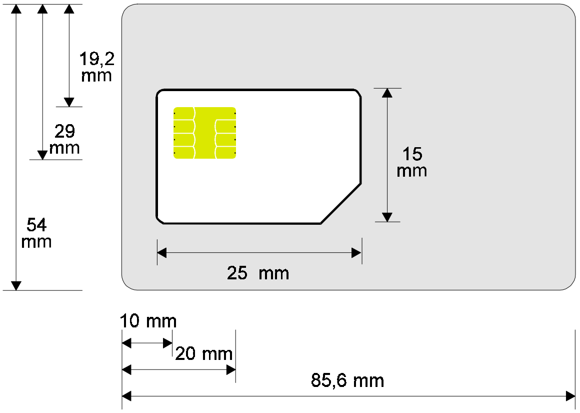


Fig. 2. Dimensions

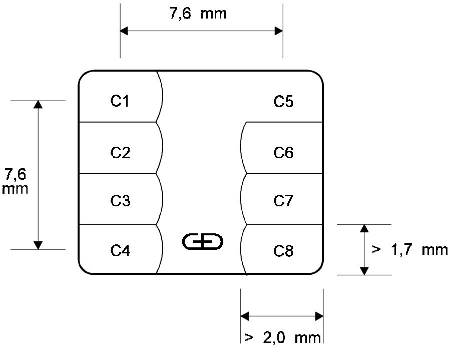
A third type, the so-called "Mini Card" or ID-00 card is 66 mm in width and 33 mm in height [1]. It has the same location of contacts with respect to the upper and the left edge as the ID-1 card. This allows the same card reader, if so designed, to accept either card. Both new formats can be thought of as obtained from an ID-1 card by cutting away excessive plastic.

2.2 Electrical Interface

Figure 3 shows a layout for a contact area providing all 8 contacts C1 to C8 specified in [11-2] (the logo of the card manufacturer is etched into the surface). The contacts C4 and C8 are reserved for future use by ISO/IEC and are often not provided.

The supply voltage Vcc is specified as 5 V ± 10% in [5]. GSM also specified a 3 Volt interface to improve battery life and thus stand-by and operation times of mobile phones [6]. Both properties have been incorporated in the revision of ISO/IEC 7816-3 [11-3]. Development within GSM has already started on mobile phones using an even lower voltage than 3 V. This will also be reflected on a new standard for GSM cards operating at 1.8 V and lower. ISO/IEC has also approved a new work item standardising low voltage interface on a general level. One of the major problems facing such specification is the task of achieving backwards compatibility between "new" and "old" phones and "old" and "new" cards. Dual voltage cards supporting 1.8 V and 3 V will most certainly not work at 5 V and may, if no precautions are taken, even be destroyed. It is expected that chips which require only 1.8 V at the interface will be available around the turn of the millennium.

Contact C2 is used for the reset signal of the card. The baudrate for the answer to reset (ATR) is equal to the frequency supplied on contact C3 divided by 372. The divisor of 372 is due to a quartz commonly used to drive the chip and which supplies about 3.57 MHz resulting in a baudrate of 9600 bits/sec. During the ATR the baudrate can (within certain limits) be increased or decreased for the subsequent communication. It is also possible to change other transmission parameters or to select a protocol with different features to the default protocol known as T=0 (see paragraph 3.2). During reset the chip shall support 1-5 MHz.



C1:	Vcc	C5:	Ground
C2:	Reset	C6:	Vpp
C3:	Clock	C7:	I/O
C4:	RFU	C8:	RFU

Fig. 3. Contact layout and assignment of the contacts

Contact C6 is used to supply the programming voltage Vpp for the non-volatile memory. An external programming voltage is needed if the chip has no internal charge pump to derive it from the supply voltage. The change from EPROM to EEPROM technology (see below) during the last years is the reason that contact C6 is no longer used.

Contact C7 is the only port for the transmission of data.

3 Performance

The performance of a smart card depends to a great extent on the chip and the protocol run between the card and the interface device.

3.1 The Microcomputer

Bearing in mind, the thickness of the card (0.76 ± 0.08 mm [8]) and the torsion and bending it will be subjected to during its life, only special purpose chips can be used. Today's microcomputers are single chip solutions which means that all the parts shown in figure 4 are integrated onto a single piece of silicon. This has some obvious advantages from the points of embedding, reliability and security. Connecting the chip to the contact area (the bonding of the chip) and embedding this module (or stamp) into the plastic carrier is a specialised profession.

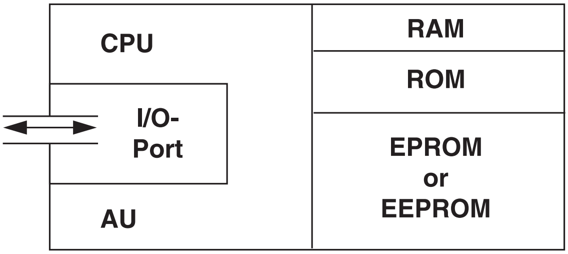


Fig. 4. A smart card microcomputer

The Central Processing Unit (CPU) comprises an 8 bit controller and is in most instances a variant of a 6805 (e.g. Motorola, SGS-Thomson), a 8051 processor (e.g. Philips, Siemens) or a manufacturer specific CPU (e.g. Hitachi). As memory is sparse, programming is usually done in assembler taking into account the specific properties of the CPU, its instruction set and the available memory. As an example let us consider the Data Encryption Standard (DES) [2] which is often used for message or entity authentication (i.e. the authentication of the card to the system or vice versa). This algorithm requires approximately 800 bytes of memory, giving a processing time of about 10 msec at 5 MHz for a 64 bit block. In the last few years the programming language C is often used if the application does not require too much memory (the overhead of programming in C is estimated at 30 %). It should be noted that many card manufacturer, who in most cases also do the software development, have their own operating system(s).

The *Read Only Memory* (ROM) is mask programmed. The photographic mask containing the customer specific ROM code is one of several layers used in the production process of the chip (see [13] for details). This has some important consequences for the design and the quality control of the software. The ROM code should only contain programs and data which are not specific to a card but are the same for a large number of cards and which are constant during the life of the card. A change of the ROM code requires a new mask. It takes several months from the completion of the software to the first die becoming available.

The ROM code typically contains the operating system of the card, the transmission protocol(s) and commands, the security algorithms and the software for the application. The card specific (secret) keys needed for the execution of the security functions are contained in the non-volatile memory.

The *Random Access Memory* (RAM) is a volatile memory the contents of which are lost when the power supply is switched off. It is used by the CPU as a buffer for storing transmission data and as a very fast access memory for the intermediate results (workspace) produced during the execution of an algorithm. Reading or writing a byte takes a few microseconds, which is a magnitude of 1,000 times faster than writing a byte into the (E)EPROM.

The non-volatile programmable memory of present-day cards consists of *Electrically Erasable Programmable Read Only Memory* (EEPROM). This type of memory allows a minimum of 100.000 update (i.e. erase/write) cycles. They derive internally the required programming voltage of about 18 V from the supply voltage Vcc. First generation smart cards had *Erasable Programmable ROM* (EPROM) with an external source for the programming voltage Vpp. As the contents of the EPROM can only be erased by UV light, every cell can be programmed just once by the CPU. The use of EPROM cards was thus limited to applications which do not require a frequent update of the memory.

One important boundary condition is the surface area of the chip which should not exceed 25 square millimetres. This and the present day technology (0.6 to 1.2 μm HCMOS) explain the size of the memory available for smart card chips (table 1).

Memory	typical	maximum	factor chip area
ROM	8 - 16 kByte	32 kByte	1
EEPROM	2 - 8 kByte	16 kByte	4
RAM	128 - 256 Byte	512 - 680 Byte	16

Table 1. Memory size

The "factor" gives a rough estimate for the silicon area used by such a cell of memory compared with the area needed for a ROM cell. This explains why RAM is sparse.

New non-volatile technologies such as Flash EEPROM and Ferro Electrical RAM (FRAM or FeRAM) will have quite an impact on the performance of smart cards. Flash EEPROM uses less area than EEPROM but can only be updated a few thousand times. Flash memory provides a good alternative to ROM, especially for the rapid production of application prototypes or different versions of operating systems. FRAM is probably the more exciting one because it has a write cycle similar to the RAM (200 ns) and maintains stored data for the same duration as EEPROM (up to 10 years). The features of these three types of non-volatile memory are shown in table 2.

The non-volatile memory is organised in so-called pages which are physically group memory cells. A page usually consists of between 4 and 64 bytes which can programmed (write/erase) in "parallel". This substantially reduces the programming time. In most cases it is also possible to program a specific byte of a page.

Technology	Write/Erase Cycles	Write / Write+Erase Time
EEPROM	10^4-10^6	1.75ms/3.5 ms
Flash EEPROM	10^3-10^4	1.5ms/3.5ms
FRAM	10^{10}	200ns

Table 2. Features of Memory Technologies

Added Units (AU) provide special functions which could otherwise not be handled at all or at least not within an acceptable time. A typical example is a coprocessor in the form of extra hardware for the execution of modular arithmetic. This is, for instance, needed for the execution of public key algorithms. Other examples of AUs are timers, Universal Asynchronous Receiver/Transmitter (UART), Memory Management Units (MMU) and hardware security logic.

3.2 The Transmission

For the exchange of data between a smart card and an interface device two protocols have been standardised [11-3]. They are denoted by T=0 and T=1, respectively. Both of them are asynchronous, half duplex protocols. The main difference between the two lies in their handling of data and the OSI reference model.

The character transmission protocol T=0 [11-3] can be characterised as a (byte-oriented) protocol of the first generation when computing power and RAM of the chips were fairly limited. Checking the parity of a byte immediately after having received it and requesting its retransmission is the only error correction. As this can not be achieved by 'standard' hardware a special UART is needed in the IFD. There is no clear separation of the transport and application layer which, for instance, makes it impossible to encrypt the header of a command. Nor is it possible to transmit data in both the request and the response of one command. This causes some overhead, for instance, during any authentication process, since data has to be exchanged between the card and the interface device. A fair amount of overhead is also unavoidable if the data to be transmitted is larger than the buffer in the RAM. The data have to be sent byte after byte until the buffer can cope with all remaining bytes. These can then be requested by using a special acknowledgement for the last byte received prior to this.

The byte-oriented transmission protocol T=0 is, however, less complex than the block protocol T=1. The standardisation of the latter was finalised in 1992. T=1 respects the OSI (Open System Interconnection) reference model and data may be sent in both a request and the response. As its name suggests, data can be handled (and transmitted) in "blocks" and the error check is carried out on a block of data.

3.3 The Throughput

Optimising an algorithm for a smart card is an act of balancing speed against memory. For the speed of the algorithm is only one of several factors which determine the performance of the card. Input, calculation and output are sequential operations. Each byte of user data is sandwiched by a start bit preceding it and by a parity bit and two stop bits following it. So the transmission of each user byte requires the transmission of 12 bits. This yields a throughput of at most 3200 user bits per second at a baudrate of 9.600 bits/sec and an infinitely fast algorithm. This does not take into account any overhead caused by the transmission protocol or the buffering of data.

As an example consider the DES, which acts on blocks of 8 bytes. Each block requires the transmission of 96 bits which takes 10 msec each way. Table 3 gives some (theoretical) values for the throughput of the card depending on the speed of the algorithm (given in both msec/DES block and bits/sec).

algorithm	40	20	10	5	msec/block
algorithm	1.600	3.200	6.400	12.800	bits/sec
smart card	1.066	1.600	2.132	2.560	bits/sec

Table 3. Throughput

One can see from table 3 that the speed of the algorithm has only a minor effect on the throughput once the transmission of a block takes about as much time as the calculation. Other means to increase the throughput are a higher baudrate (e.g. 115 kbit/sec) and the use of only one stop bit, which is possible in T=1.

4 Smart Card Operating System

Operating systems are used on different digital hardware platforms – from smart cards, pocket calculators to organisers, PCs and mainframes. Broadly speaking, the term operating system, for which there is no general definition, refers to a group of system programs required for operating a computer.

The operating system provides defined functions to the user, who in turn need not know anything about the computer hardware. The user can run and program applications almost completely independently of the hardware. The operating system takes care of controlling and organising memory media (RAM, hard disks, CDs, etc.) and of programming processors. It allows the hardware manufacturer to incorporate many technological developments without there being a need to change functions of the operating system.

What has been said about operating systems for "large-scale" computers also applies to operating systems for smart cards, the ultimate personal computer. A smart card operating system and its basic tasks can be characterised as follows:

- miniature operating systems requiring memory capacity of a few kBytes;
- administration of security hardware based on single-chip microcontrollers;
- single processing systems;
- machine-interfaces in the form of a serial interface;
- secure data storage;
- support of cryptographic methods for authenticating system components and enciphering/ deciphering data;
- provision of IT-security for applications.

The operating system and the smart card hardware have to protect themselves against the following three basic threats for the duration of their entire life cycle (see 4.4 below):

- loss of confidentiality by spoofing or unauthorised information about programs and data such as keys, PINs and user data;
- loss of integrity by manipulation or unauthorised modification of information such as identification number and counters of electronic purses;
- loss of availability by unauthorised withholding of information or system functions.

4.1 Software Standardisation

The basis for all industry specifications and thus the most important smart card standard is "ISO/IEC 7816, Identification cards – Integrated circuit(s) cards with contacts" [11]. The following list describes the major topics of some parts of this International Standard which are relevant for smart card operating systems:

- ISO/IEC 7816-3: Transport protocols T=0 and T=1;
- ISO/IEC 7816-4: File organisation, commands, historical bytes and secure messaging;
- ISO/IEC 7816-5: Definition and registration of the Application IDentifier (AID);
- ISO/IEC 7816-6: Definition of data objects;
- ISO/IEC 7816-7 (Draft): Commands for Structured Card Query Language (SCQL) for database application;
- ISO/IEC 7816-8 (Draft): Security functions for Public key procedures and extended PIN functions;
- ISO/IEC 7816-9 (Draft): Personalisation commands and extended file handling commands;
- ISO/IEC 7816-11 (Draft): Access conditions and security attributes.

Even if smart card operating systems have been implemented in accordance with these International Standards, it does not mean that they are compatible with each other. This is to some extent due to the options provided for in the standards.

Specifications are thus required to select particular items from the standard to implement compatible smart card operating systems (of different manufacturers) for one and the same application. One such specification is GSM 11.11. Furthermore, some applications as, for instance, electronic purses require functionality not defined in ISO/IEC 7816. They need, therefore, "private-use-commands". Today, approximately 60 % of the functions of a smart card operating system are private use. In 1994 the leading credit card organisations EUROPAY, MasterCard and VISA (EMV) began to draft a specification for smart cards in international payment systems based on ISO/IEC 7816. In June 1996, version 3 of the EMV specifications has been published [3]. This allows the implementation of the basic command set for compatible credit card systems based on smart cards.

4.2 Structure of Smart Card Operating Systems

The flow of information between an interface device and a smart card occurs via transport protocols in the form of command-response pairs. In most cases the interface device or the application (terminal, PC, etc.) has the role of the master, i.e. the commands will be generated and processed by the IFD. In those cases, where the smart card represents the application (e.g. in the case of security modules or in specific applications within the GSM SIM Application Toolkit [7]), the roles may be reversed (see figure 5).

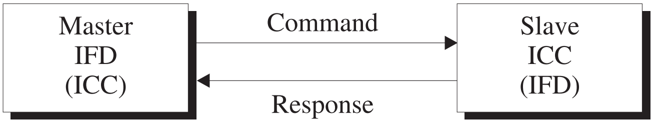


Fig. 5. Information transaction

In accordance with the OSI 7-layer model the information transaction can be divided into three protocol sections:

- physical layer (layer 1);
- data transmission protocols (layer 2);
- information protocols, for command and response data (layer 7).

Smart card operating systems are divided into modular functional units called managers – just like computer operating systems. According to reference [18] an information transaction may run through the following managers (see figure 6):

- The *Transport Manager* controls and secures the data transmission using asynchronous, half duplex transport protocols. These can be the already mentioned protocols (T=0, T=1) or a national protocol (all national protocols no matter how different they may be are denoted by T=14).
- The *Secure Messaging Manager* covers the cryptographic protection of the transmission by enciphering or deciphering information and/or by checking information for authenticity.

- The *Logical Channel Manager* is required for accessing an application if several applications are "open" simultaneously. This semi-multitasking is required, for example, in the case of a bank application transferring money to an electronic purse.
- The *Command Manager* verifies command syntax. In some operating systems it also controls the command processing protocol by using the state machine(s) of an application.
- The *Security Manager* is in charge of object access control, in particular for keys. It controls, for instance, the states of the state machine which are depending on the successful or unsuccessful execution of the identification of the user and of the authentication mechanisms.
- The *File Manager* administers the different file categories and supports the different file types.
- The *Memory Manager* administers the entire memory organisation, e.g. installation, applications and files. It calculates checksums, repairs defective file structures and takes care of the administration of the available memory.
- *Functions*. Among these are, for example, mathematical operations for specific cryptographic functions. In some chips they are supported by hardware. These functions may be part of the whole system or just part of one or more applications.

The smart card operating systems of the STARCOS® family (**Sm**ar**t** **C**ar**d** **C**hip **O**perating **S**ystem) [14] incorporate these managers as part of the system concept. Since these universal, application-independent operating systems offer structured administration they are suitable for administering several applications and issuers on one card.

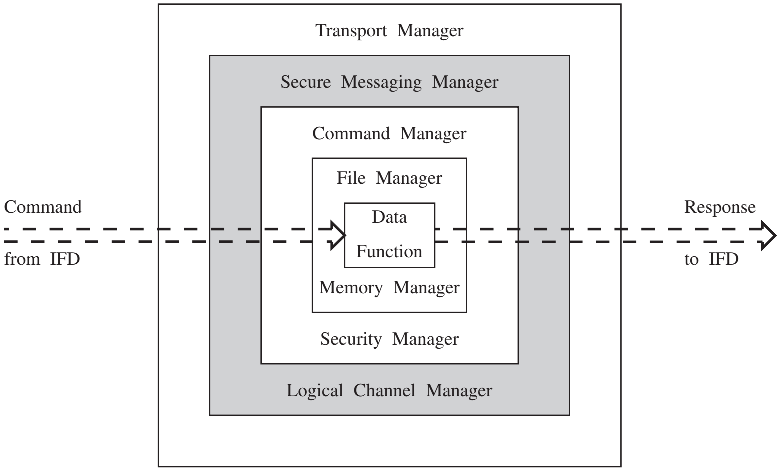


Fig. 6. Structure of a smart card operating system

4.2.1 File Organisation

A smart card operating system administers files in directories – similar to conventional PC operating systems. One distinguishes the following components within the file organisation which one can visualise as having a "tree-structure" (figure 7):

- Master File (MF): the root of the file system (tree);
- Dedicated Files (DF): a DF contains substructures and the actual application (DFs are also called "directories");
- Elementary Files (EF): an EF contains application information or data.

The MF is usually selected implicitly after the card has been reset. It may also contain an application, for instance on a mono-application card. In the case of multiapplication cards each application is contained in a separate dedicated file or in further subdirectories. An application is selected by using an application identifier (AID), which can be registered on both international and national level with ISO/IEC [11-5], or by a (fixed) file identifier consisting of two bytes. Application identifiers have the advantage that the interface device does not have to know the file identifier for the application and that, as a registered AID (a so-called RID) is unique world-wide, several applications can easily be combined on a card. In the case of (fixed) file identifiers a collision can arise if two applications which use the same file identifier are to be combined on one card. Data which is used for all applications in the card (for example administrative and general security information such as serial number, keys, PIN) and data concerning the administration of the card life cycle are stored in the master file. This information is, for instance, used by the operating system for the creation of a new application. Application specific control information and files are stored in a DF containing the application. They are separated logically and, in some cases, physically from other applications contained in different DFs. The administration of rights for DFs arranged in a "vertical" structure is always performed by the "parent" DF i.e., the DF which resides "one level up". A dedicated file can administer several dedicated files which are "one level below". An example would be a financial card where the same service provider is responsible for a payment transaction application and credit, debit and electronic purse functions.

ISO/IEC 7816-4 [11-4] distinguishes two categories of Elementary Files (EFs). While the data contained in a Working Elementary File (WEF) cannot be interpreted by the operating system, the data stored in a Secret File (ISF) must be interpreted by the operating system. A DF may contain more than one WEF but only one ISF. Typical information stored in an ISF are keys which are to be protected from read by the outside world. Smart card operating systems such as STARCOS® may, in addition, support files which are of "mixed" type, i.e. some of their data can be interpreted while the remaining data cannot be interpreted by the operating system.

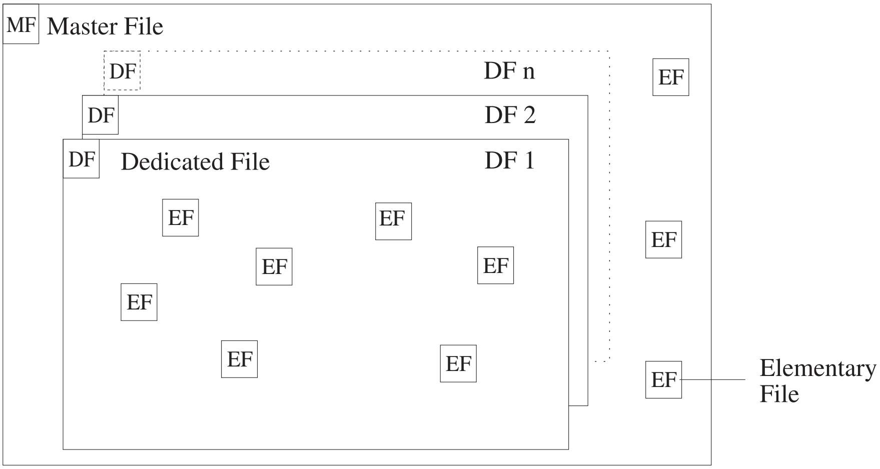


Fig. 7. File organisation

4.2.2 File Structures

The structure of an EF depends on its use. The following four types are defined in ISO/IEC 7816-4 [11-4]:

- *transparent files* consisting of a sequence of bytes having an amorphous structure;
- *linear fixed files* consisting of records having the same length;
- *linear variable files* consisting of records of variable length;
- *cyclic files* having records of fixed length which are organised in a ring structure, the oldest entry will be overwritten by the entry to be stored.

The four basic structures can be extended by other file structures. Examples are database applications (database files), electronic purses (compute files) or public key applications (internal public key files).

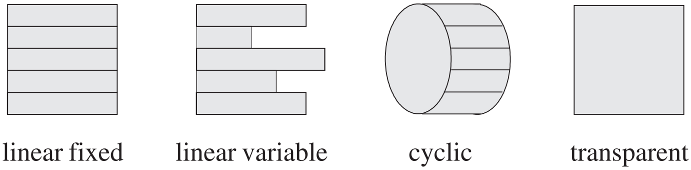


Fig. 8. Data structures of elementary files

The definition of the category, the type and the access possibilities (read, write, update, delete,...) with their respective access conditions (never, always, PIN, special authentication,...) are stored in the file header of each file.

4.2.3 Commands

The functionality of an operating system is not only reflected in the number of available commands but also in their complexity. This grows with the need for more security for an application. The commands defined in ISO/IEC 7816-4 [11-4] are the *basic* command set and can be grouped in the following functional classes:

- file selection;
- read data;
- modify and delete data;
- generate data;
- compare data;
- authenticate using cryptographic functions.

Due to the complexity of today's applications approximately 60% of the commands of an operating system are "private-use-commands" and not defined in [11-4]. Examples are commands for mutual authentication and cryptographically secured counter functions representing chained basic functions, so-called macros.

4.3 Security

Operating systems quite often administer applications with high security requirements. The security of a smart card is a combination of the security of the chip (hardware) and the operating system (software). To obtain optimum security the programming of an operating system requires extensive knowledge of the properties of the hardware in particular with respect to electrical characteristics, detectors, interrupts, timing and other features which may influence the security. From a software point of view one can distinguish between functional security and security against manipulations.

Functional security can be guaranteed by:

- transport protocol;
- command interpreter;
- file organisation, file structure, data objects;
- functions;
- layer separation;
- error detection and correction functions.

Security against manipulation can be obtained by implementing:

- secure messaging;
- identification and authentication;
- state machines;
- object protection;
- digital signature;
- proof retention;
- random numbers generators.

4.4 Card Life Cycle

The operating system contributes considerably to the guaranteed card life cycle of a smart card. The card life cycle – from production to deactivation of the card (see figure 9) – is divided into the following five phases according to ISO 10202-1 [12]:

- Phase 1 – chip and card manufacturing
  development of the operating system and transport to chip manufacturer;
  implementation of the operating system, usually as a ROM mask;
  chip production and transport to card manufacturer.
- Phase 2 – card preparation
  initialisation and prepersonalisation of the card, i.e. loading constants and system-related data;
  dispatching the cards to the issuers.
- Phase 3 – application preparation
  assignment, personalisation and activation of one or several applications.
- Phase 4 – application phase
  using global card functions and application access;
  using management functions (administration) of applications (lock, release).
- Phase 5 – termination of use
  delete keys and the complete application.

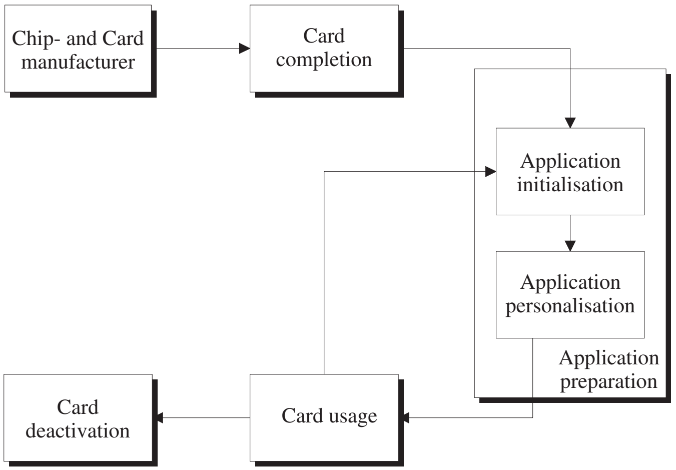

Chip- and Card manufacturer
Card completion
Application initialisation
Application personalisation
Application preparation
Card deactivation
Card usage

Fig. 9. Card life cycle

5 Applications

The main security functions of a smart card are the identification of the user, the authentication of the card by the system and, depending on the application, the authentication of the system by the card. This involves the secure storing of secret information as well as the secure execution of cryptographic algorithms.

The identification of a user is usually done by means of a Personal Identification Number (PIN). The PIN is verified by the microcomputer of the card which compares in its RAM the PIN presented with the PIN stored. If the comparison is negative, the CPU will refuse to work. The chip also keeps track of the number of consecutive wrong PIN entries. If this number reaches a pre-set (card specific) threshold (usually three), the card blocks itself against any further use. The change of the PIN by the user, which is a standard feature of smart cards, may be subjected to the exclusion of trivial values such as "0000", "123456", "4711", "0815" or the date of birth of the user (if known) which could be stored in the card.

At the beginning of any session is authentication, the "corroboration that an entity is the one claimed" (ISO/IEC 9798-1). Entity authentication is achieved by using a "challenge-response method" employing a cryptographic algorithm such as the DES or a public key function. To authenticate the card the system challenges the card by sending a random number. The card uses this and its own card specific key as input to its cryptographic algorithm. The output of the calculation, the response, is transmitted to the system. This compares the value received with the one calculated itself. If the two match the card is considered to be genuine. Analogously the card can check the authenticity of the system.

5.1 GSM

The Global System for Mobile communications (GSM) is a digital, cellular radio system with more than 170 networks on air in about 100 countries. Access to all these networks is controlled by smart cards, the so-called Subscriber Identity Modules (SIMs), in form of ID-1 or Plug-in cards. GSM is the first world-wide application based on smart cards and has given a huge impetus to their standardisation and use.

Subject only to their size and roaming agreements between the network operators all SIMs work in all mobile phones in all networks. The general properties of the interface between the SIM and the mobile phone are laid down in GSM 11.11 [5] while GSM 11.12 [6] was the first specification for a 3V card interface. The latest specification is GSM 11.14 [7] the so-called SIM Application Toolkit. This provides the operators with a toolbox to create their own applications on the card. Information containing commands and data may be downloaded over the air into the SIM, transparently to the mobile phone. After interpreting the information, the SIM will react as requested by, for instance, updating a file or creating a new file or even a new application. The SIM may also become pro-active and request the mobile phone to act on its behalf. For more information on GSM and SIMs the reader is referred to [16].

5.2 Company ID-Cards

Using a smart card as a company ID-card and, at the same time, for access control to the company computer network can solve quite a few security and handling problems arising from passwords and access control in general.

The communication between the card and the system can be protected by message authentication codes to secure the update of sensitive data in the card. The combination of a smart (company) card with PIN has several advantages over Password and User-ID. Access to the network requires the possession of the card *and* the knowledge of the PIN. Knowing just the PIN is not sufficient. If the holder "lends" the card to someone else this will be temporary. Without the card the holder may not be able to leave or enter the premises or parts thereof. Passwords can be passed on indiscriminately; company cards will not. A proper log-off can be enforced by making the entry to the (computer) room depend on the card and by checking the presence of the card, creating security entries if the card is forcefully removed.

To enhance the security, visual information can be engraved into the card body for example by means of a laser beam. This information could consist of the usual data such as date of issue and card number as well as typical data of an ID-card such as the photograph and the signature of the card holder.

5.3 Banking Cards

The first nation-wide smart card application was a banking card in France. The field-test using memory chips took place in Lyon in 1983. Shortly afterwards microcontrollers from Motorola with EPROM as non-volatile memory were introduced at national level. The transition from magnetic stripe to microcontroller with on-board memory had the following advantages from a security point of view:

- administration, storage and validity check of the PIN by and in the chip;
- authentication and enciphering of data using a cryptographic algorithm.

Smart cards supporting electronic purse applications were introduced on a national level in Austria in 1995 and in Germany in 1996. All these cards, which were issued as a replacement for the eurocheque cards with magnetic stripe, were hybrid cards with both a magnetic stripe and a microcomputer. Apart from the usual Point of Sale (POS) functions of the magnetic stripe the chip supports an electronic purse. Since the credit limit is also administered by the chip POS transactions can now be handled off-line. As in most electronic purse systems (e.g. Proton, VISA Cash, STARCOIN) the loading of the purse is done on-line while purchasing can be done off-line. With the availability of on-board hardware units for special arithmetic the symmetric cryptoalgorithms such as DES are expected to be replaced with asymmetric (public key) algorithms in the coming few years. This will significantly improve both the key management and the cryptographic protection.

5.4 Other Applications

Not only banking cards will benefit from the introduction of digital signatures and authentication mechanisms using public key based cryptosystems. The recent development of such systems and the improvement of the added hardware units now allow the computation of a digital signature employing a key of 512 to 1024 bits in much less than 1 second. This is of particular importance for the security of information and transactions for applications in world-wide network.

Contactless cards are supposed to be ideal for applications where the cards of a large number of people may have to be handled in a very short time. A typical example is a "city card" with an electronic purse for the combined use of (public) transport and public amenities such as libraries, swimming pools and museums. An example of a hybrid card in both meanings of this term is the Lufthansa chip card. Its magnetic stripe (and embossing) serve as a credit card, its contactless microcontroller chip is used for electronic ticketing and boarding, and the memory chip with contacts as a subscriber card for the public German telephone system.

Another large application are the so-called health cards which may contain a database of the medical history, allergies and other relevant information such as address and insurance number of the card holder. Access and update rights depend very much on the data and may be divided between the patient and the respective medical doctors. Access is defined on the level of data objects and not on files. A surgeon, for instance, may thus only access data relevant for the diagnosis and treatment of cancer.

6 Standardisation

Though chip cards are quite standardised objects this does not mean that there is nothing left to do. Conformance testing and common test methods are one subject. A standard, solely dedicated to test methods, is ISO/IEC 10373 (1993). This is one of the many topics within the scope of SC 17, a subcommittee of the Joint Technical Committee 1 (Information technology) of ISO and IEC. SC 17 is also responsible for the International Standard ISO/IEC 7816 and other basic International Standards for identification cards. One of its Working Groups (WG) develops a multipart standard for contactless cards ISO/IEC 10536. The first three parts of this standard correspond to the respective parts of ISO/IEC 7816. The work on proximity contactless cards (ISO/IEC 14443) has just started.

Another international standardisation body dealing with smart cards is ISO/TC 68 "Banking and related financial services". It edits two multipart standards: *Messages between the integrated circuit card and the card accepting device* (ISO 9992) and *Security architectures of financial transaction systems using integrated circuit cards* (ISO 10202).

Within Europe chip cards are standardised by CEN, the Comité Européen de Normalisation, and by ETSI, the European Telecommunications Standards Institute. The Technical Committee TC 224 of CEN has 13 working groups which specify items from physical characteristics (WG 1) to airline applications (WG 14). Its WG 9 was closely related to the former ETSI SubTechnical Committee STC TE9 which produced a standard on application independent card requirements for telecommunications [4]. This specified for the first time commands which went beyond the basic erase, read, write and update functions. This is now to a large extent superseded by part 4 and the draft versions of parts 7 and 8 of ISO/IEC 7816. WG 10 of TC 224 develops standards for electronic purse systems. ETSI produced several smart card specifications for specific applications. The most important ones are probably those written for GSM. Others are the DECT Authentication Module (DAM) specification for the Digital Enhanced Cordless Telecommunications and its test specification. ETSI has now formed a new Technical Committee to write a generic core specification for telecommunication cards.

More information on the committees mentioned can be obtained from the national standards institute of the relevant country.

Apart from the standards produced and developed by regional and international standards organisations there exist quite a few "industry standards". The most important one of these is the so-called EMV-specification for payment systems which has been written jointly by EUROPAY, MasterCard and VISA [3]. The three parts define the IC Cards, the card terminal and the card application.

7 Outlook

The advancement of the semiconductor technology in the last years had a dramatic effect on the memory provided by smart card chips. While 5 years ago chips providing 8 kByte of ROM, 128 Byte of RAM and 3 kByte of EEPROM were the flagships of all manufacturers, it can be assumed that in a few years time semiconductor technology will have a characteristic distance below 0.6 m and high end microcomputers will have in excess of 64 kByte of ROM and 32 kByte of EEPROM. The values given in table 1 as maximum will be part of the typical range within the next year. This and the progress of the tools will have a strong influence on the development of operating systems and applications. It will become common to write most of the ROM code not only in C but also in the form of modular blocks. This will immensely improve the portability of the software from one microprocessor to another, even if they have "incompatible" CPUs, as well as the design of true multiapplication operating systems, which require in excess of 10 kByte of ROM without the coding of the application.

As the applications (to be) supported by a smart card become more and more complex, today's CPUs with their CISC (Complex Instruction Set Computer) architecture and their 8 bit controllers, which are by comparison with other areas quite outdated, need to be improved. Several chip manufacturers have started extending the cores of their microcomputers by introducing additional instructions and ways of addressing memory as well as higher clock rates (e.g. 10 MHz). This upgrading has the advantage that existing software is supported and can easily be ported. There is, however, a limit to the improvement of the performance. This can only be achieved on a major scale by replacing the core by a RISC-CPU (Reduced Instruction Set Computer). This architecture will, for instance, allow internal frequencies of 35 MHz and higher. It will also further the introduction of high level languages. Several manufacturers have already announced their intention to introduce this architecture which has been widely used in other areas for several years (figure 10).

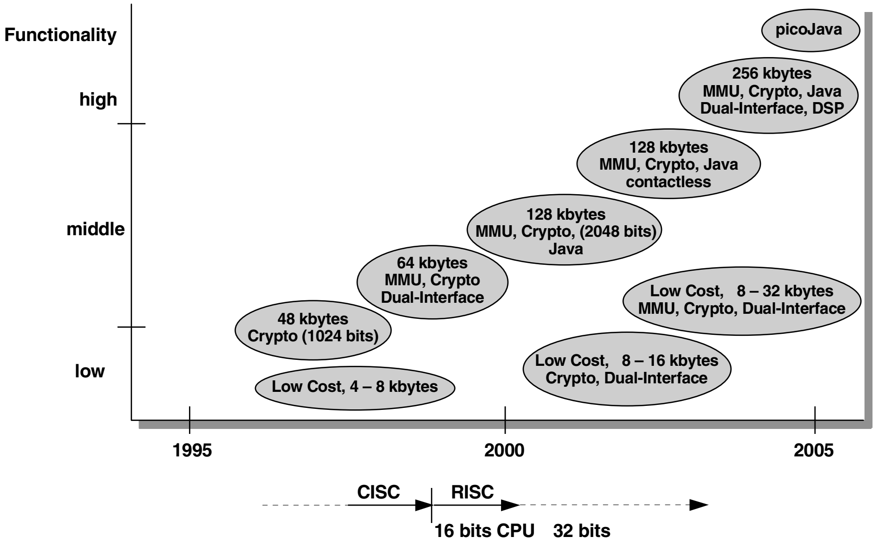


Fig. 10. Trends of smart card microcontroller

Larger microcomputers with an improved performance will increase the potential for multiapplication cards and the requirement to download applications after the card has been issued to the user. When loading a new application into a card several issues have to be considered very closely. Is the new application independent of the hardware (CPU and memory) of the chip and the implementation of the operating system? How does it interact with the resident application(s)? Can security problems definitely be excluded? For these reasons the loading of another application with *executable code* needs to be evaluated by the software engineers who are responsible for the operating system and the resident applications. To ease this process and to achieve the goal of downloading applications into issued cards Memory Management Units (MMU) and Interpreter Languages will be used. Memory Management Units support the operating system in controlling the memory (firewall function) and allow the use of the CPU by the downloaded application with executable code. Existing interpreter solutions are mostly proprietary and slow down the execution of applications and programs.

In 1995 SUN Microsystems presented Java, an interpreter language for a large variety of microprocessor platforms. It was originally intended for the linkage of set-top-boxes, copiers and other electronic consumer goods with a microprocessor. Since

Java was provided free of charge on the Internet and was running on such a variety of hardware platforms it immediately became the language of the Internet. Java has completely realised the concept of object orientation and provides security mechanisms which make it suitable also for smart cards. The compiler produces a byte code which is interpreted by a virtual machine, the so-called Java Virtual Machine (JVM). The functionality of this machine is independent of the processor and can, by means of the Java API (Application Program Interface), access the operating system of the smart card. It can thus also make use of hardware dependent functions such as some security, utility and I/O routines (see figure 11). The standardisation of the Java byte code, the JVM and the JC (Java Card) API for smart card is progressed with great determination by SUN and several smart card manufacturers. As with all interpreter solutions Java has, however, an adverse effect on the performance of the microprocessor. Several chip manufacturers have announced the design of new hardware to support the JVM. These products, which will greatly improve the performance, are expected to be available within the next couple of years [17]. Complete solutions, i.e. the JVM in hardware (pico Java) for smart cards, will probably not be available in this millennium.

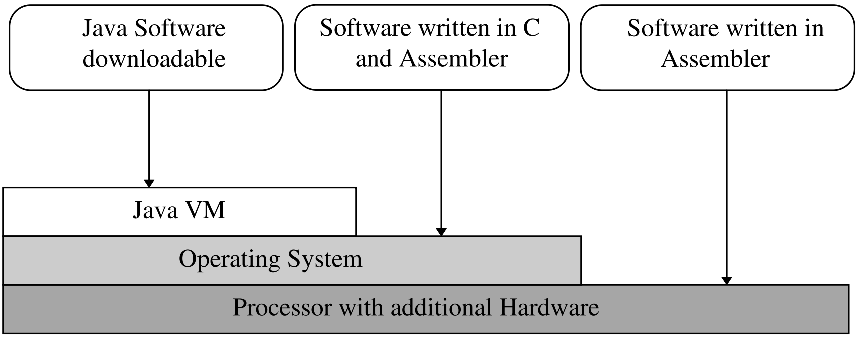

Java Software downloadable
Software written in C and Assembler
Software written in Assembler
Java VM
Operating System
Processor with additional Hardware

Fig. 11. Software designer's view

All present Java implementations on smart cards make use of a two-part Java virtual machine (JVM) consisting of an Off-Card JVM in a PC and an On-Card JVM on the card itself. The Off-card JVM does the pre-processing of the "JAVA byte-code" and, in particular, the class files. This optimisation is needed for the processing of the byte code by the CPU of the smart card, since current smart card chips are not powerful enough for the efficient execution of the standard JAVA byte code. For this reason a standardisation of the downloading interface between the Off-card JVM and the On-card JVM is important to avoid the development of proprietary solutions.

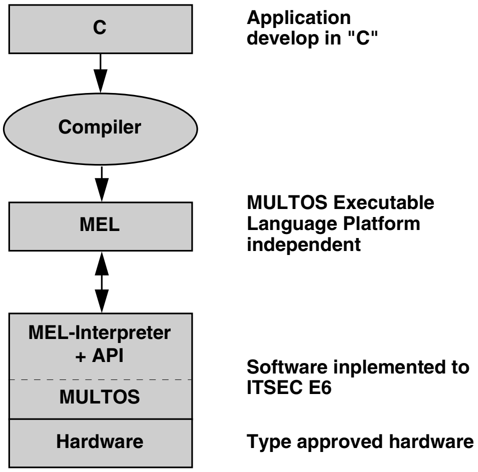


Fig. 12. Application design with MULTOS

The other new interpreter operating system is MULTOS which is supported by MasterCard and MONDEX. Similarly to Java, the core of the operating system is an interpreter which allows the application to be developed independently of the underlying hardware. Applications can be written in the high level language C and are then translated with the help of a C-compiler into the interpreter language MEL (MULTOS Executable Language). MEL is thus a specified interface for downloading software. The MEL code is interpreted by the smart card interpreter using the API (see figure 12). The security of MULTOS is expected to satisfy ITSEC E6, the highest level of the ITSEC certification. The first application to run on MULTOS is probably the electronic purse application from MONDEX. Similarly to Java, applications requiring the execution of complex or time critical (security) functions will need a special API to provide an acceptable execution time for the application. As is the case with all interpreter languages, the widespread use of this language will depend on the development of hardware specific for MULTOS. This new hardware could be an extension of the CPU with respect to the virtual machine or an additional processor dedicated to the specific interpreter.

Smart cards are also security devices and will take over more and more security related functions of the systems they are used in. The introduction of technologies of 0.6 m and below will not only allow the production of chips providing more memory, it will also provide an hitherto unknown protection against potential future attacks. Extra hardware for public key cryptography (be it RSA with a key length of 2084 bit or elliptic curve systems) will provide the means for secure (off-line) authentication and digital signatures. Digital Signal Processors (DSP) on-board a smart card microprocessor will pave the way for the usage of biometric systems for user authentication and may make the PIN obsolete.

The smart card of the future will be a PC in pocket size with sensors for biometric features and a human interface.

Glossary

ATR	Answer To Reset
CEN	Comité Européen de Normalisation
CPU	Central Processing Unit
DES	Data Encryption Standard
DF	Dedicated File
EF	Elementary File
ETSI	European Telecommunications Standards Institute
EEPROM	Electrically Erasable Programmable ROM
EPROM	Electrically Programmable ROM
FRAM	Ferro electrical RAM
GSM	Global System for Mobile communications
IC	Integrated Circuit
ICC	Integrated Circuit(s) Card
IEC	International Electrotechnical Commission
IFD	InterFace Device
ISF	Internal Secret File
ISO	International Oganization for Standardization
JVM	Java Virtual Machine
MF	Master File
OSI	Open System Interconnect(ion)
PIN	Personal Identification Number
POS	Point of Sale
RAM	Random Access Memory
ROM	Read Only Memory
UART	Universal Asynchronous Receiver/Transmitter
WEF	Working Elementary File

References

[1] CEN Draft ENV 1375-2: Identification card systems - Intersector integrated circuit(s) card additional formats - Part - 2: ID-00 Card size and physical characteristics.

[2] Data Encryption Standard (DES). Federal Information Processing Standards Publication 46, National Bureau of Standards 1977.

[3] EMV '96, Specification for Payment Systems, EUROPAY, MasterCard and VISA, version 3.0, June 1996.

[4] EN 726-3:1994, Identification card systems - Telecommunication(s) integrated circuit(s) cards and terminals - Part 3: Application independent card requirements.

[5] GSM 11.11 (ETS 300 608), Digital cellular telecommunications system (Phase 2); Specification of the Subscriber Identity Module - Mobile

Equipment (SIM-ME) interface.
GSM 11.11 (ETS 300 977), Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface.

[6] GSM 11.12 (ETS 300 641), Digital cellular telecommunications system (Phase 2); Specification of the 3 Volt Subscriber Identity Module - Mobile Equipment (SIM - ME) interface.

[7] GSM 11.14, Digital cellular telecommunications system (Phase 2+); Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface.

[8] ISO/IEC 7810 (2nd edition): 1995, Information technology - Identification cards - Physical characteristics.

[9] ISO/IEC 7811 (2nd edition): 1995, Information technology - Identification cards - Recording technique.

[10] ISO/IEC 7813 (3rd edition): 1990, Information technology - Identification cards - Financial transaction cards.

[11] ISO/IEC 7816, Information technology - Identification cards - Integrated circuit(s) cards with contacts.

Part 1: 1987, Physical characteristics. (Under review).

Part 2: 1988, Dimensions and location of the contacts. (Under review).

Part 3: 1989, Electronic signals and transmission protocols. (Under review).

Part 4: 1995, Interindustry commands for interchange.

Part 5: 1994, Numbering system and registration procedure for application identifiers.

Part 6: 1994, Interindustry data elements.

Part 7 (Draft): Interindustry commands for Structured Card Query Language (SCQL).

Part 8 (Draft): Security related interindustry commands.

Part 9 (Draft): Enhanced interindustry commands.

Part 11 (Draft): Security architecture.

[12] ISO 10202-1: 1991: Banking, securities and other financial services - Financial transaction cards: Security architecture of financial transaction systems using integrated circuit cards - Part 1: Card life cycle.

[13] M. Paterson, Secure Single Chip Microcomputer Manufacture, in: D. Chaum (ed.), Smart Card 2000, North Holland 1991, 29-37.

[14] STARCOS S2.1, Reference Manual, 10/96, Giesecke & Devrient GmbH, Munich.

[15] J. Svigals, Smart Cards. The ultimate personal computer. Mac Millan Publ. 1985.

[16] K. Vedder, GSM: Security, Services and the SIM, this volume, pp. 227-243.

[17] P. Wayner, Sun Gambles on Java Chips, in: Byte Nov. 1996, 79-88.

[18] F. Weikmann, Chipkarten - Entwicklungsstand und weitere Perspektiven, in: PIK 1/93, 28-34.